UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

MASTER IN COMPUTER SCIENCE

Pattern Recognition

Lecture 3 : Clustering

Dr. Andreas Fischer
andreas.fischer@unifr.ch

Classification vs Clustering

- Classification
  - Classes known
  - Training samples labeled with their class
  - *Supervised* learning: $f_\theta(x)=y$ based on labeled samples
  - *Semi-supervised* learning: $f_\theta(x)=y$ based on labeled samples as well as unlabeled samples (self-learning, co-learning, …)
- Clustering
  - Classes unknown
  - Unlabeled samples
  - *Unsupervised* learning: group unlabeled samples into classes based on their similarity
- In the following, three standard approaches for clustering are discussed: hierarchical clustering, k-means clustering, and graph-based clustering.

Hierarchical Clustering

Hierarchical Clustering

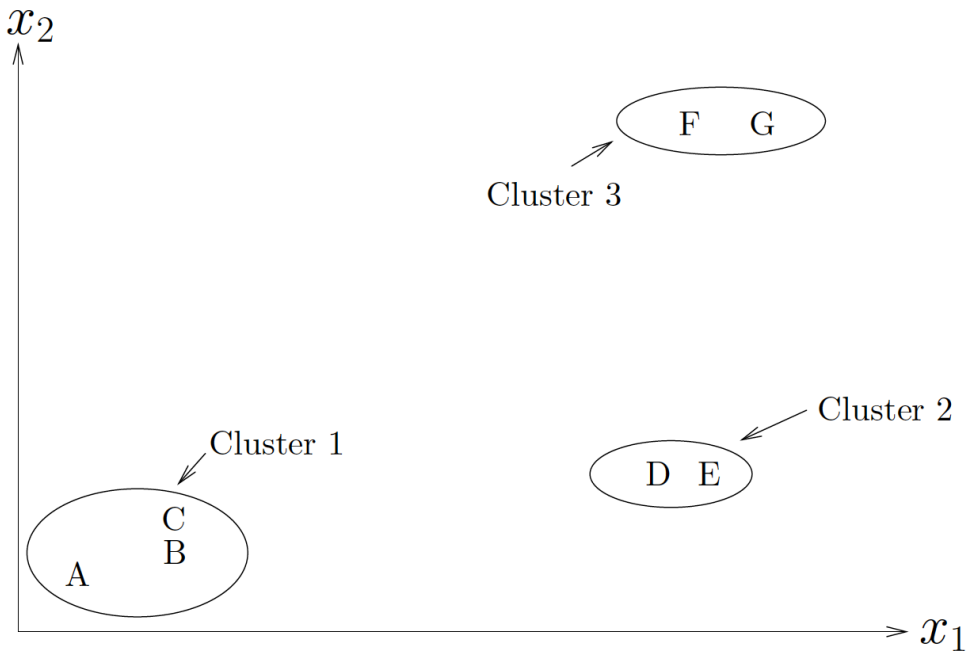
- Segment a set of samples $x_1,\ldots,x_N$ with $x_i \in R^n$ into subsets.
- *Bottom-up* agglomerative approach.
- Time complexity $O(\delta N^3)$, faster variant $O(\delta N^2 \log N)$, where δ is the complexity for computing the cluster distance $d(C_i,C_j)$.

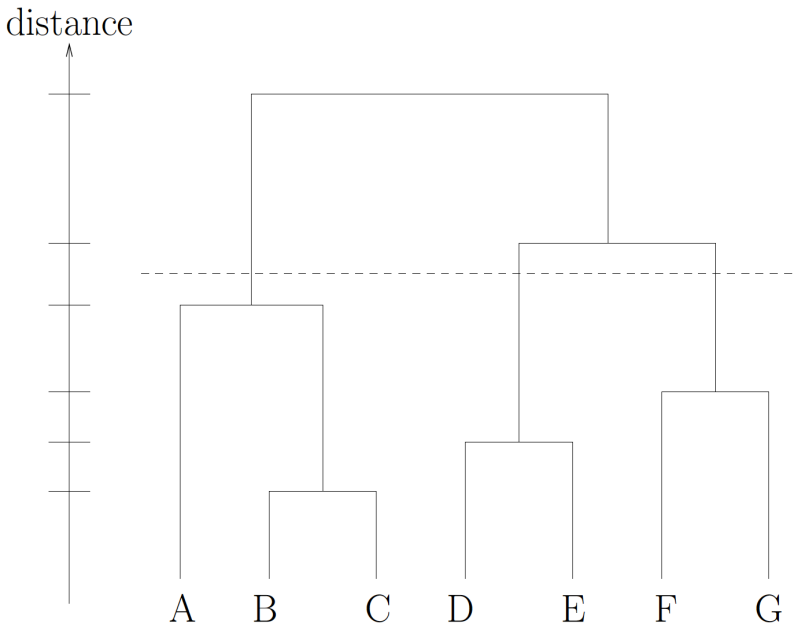
Require: training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
Ensure: hierarchical clustering solutions $R_0, \ldots, R_{N-1}$

1: $R_0 = \{C_1 = \{x_1\}, \ldots, C_N = \{x_N\}\}; t = 0$
2: **repeat**
3: $t = t + 1$; $R_t = R_{t-1}$
4: find C_i, C_j with $d(C_i, C_j) = \min\{d(C_r, C_s) | C_r \neq C_s; C_r, C_s \in R_t\}$
5: define new cluster $C_q = C_i \cup C_j$
6: remove C_i and C_j from R_t
7: add C_q to R_t
8: **until** all x_i belong to the same cluster C

Example

- Resulting clusters can be represented by a tree with N levels.
- Each level is attributed with the corresponding cluster distance $d(C_i,C_j)$.
- Such a tree is also called *dendrogram*.





Cluster Distance

- *Single-linkage* distance:

$$d(C_i, C_j) = \min\{d(x, y) \mid x \in C_i, y \in C_j\}$$

- *Complete-linkage* distance:

$$d(C_i, C_j) = \max\{d(x, y) \mid x \in C_i, y \in C_j\}$$

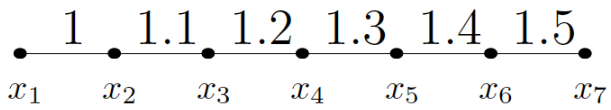
- *Average* distance:

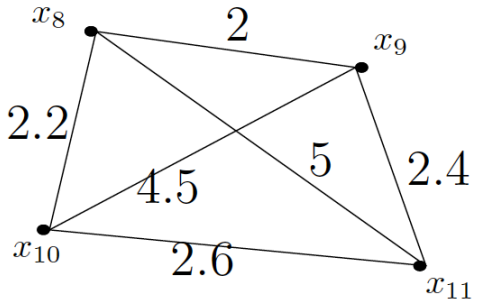
$$d(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$

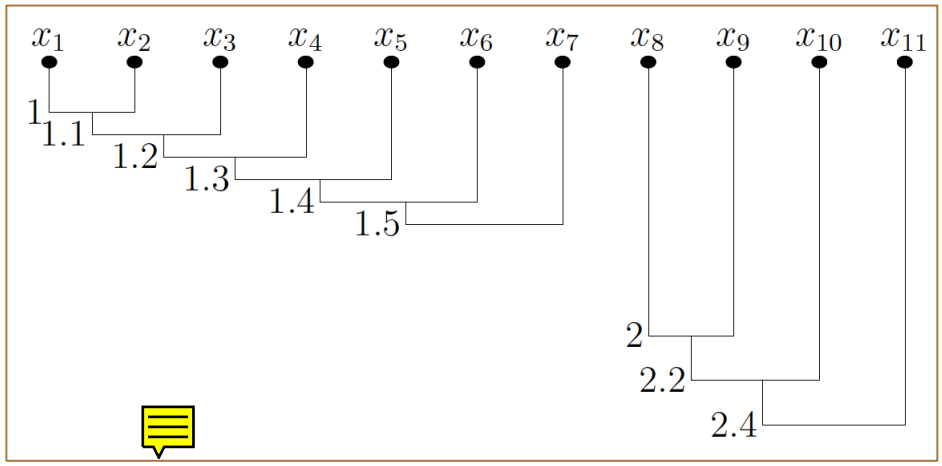
- Clusters built with complete-linkage tend to be more compact than those built with single-linkage. The average distance is a trade-off in-between.
- Another possibility would be to define the cluster distance as the distance between the mean vectors.

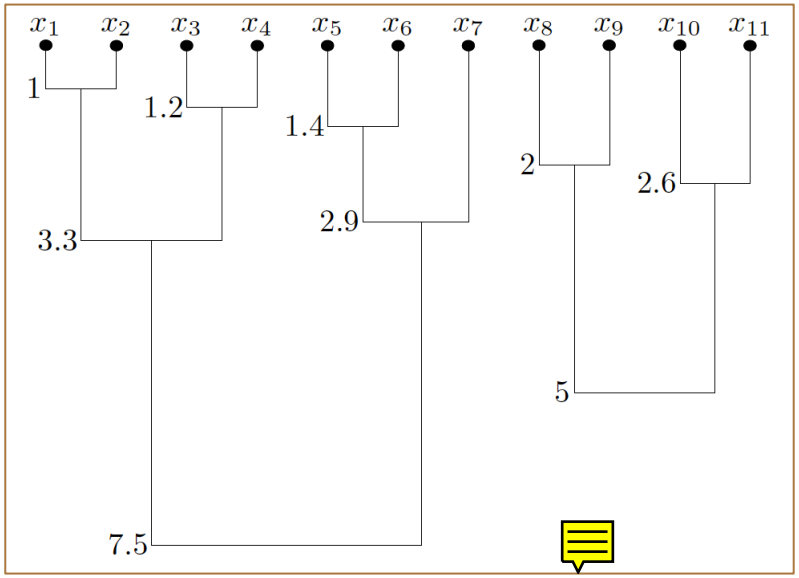
Example

- Single-linkage vs complete-linkage.
- The two groups of patterns $\{x_1,\ldots,x_7\}$ and $\{x_8,\ldots,x_{11}\}$ are considered to be far apart.









K-Means Clustering

K-Means

- One of the most widely used algorithm for clustering.
- Finds exactly K clusters, that is the number of clusters has to be known in advance.

Require: training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; number of clusters K
Ensure: clustering $C = \{C_1, \ldots, C_K\}$

1: choose K initial cluster centers $m_1, \ldots, m_K$
2: **repeat**
3: assign every x_i to the cluster with the nearest center m_j
4: recompute m_j for each cluster
5: **until** termination criterion
6: (optional post-processing)

K-Means

- Possible termination criteria:
  - Fixed number of iterations
  - Small or no change of the cluster centers
  - Small or no change of the cluster assignment
  - Small clustering error E_K (see next slide) or small decrease of E_K
- Possible choices of initial cluster centers:
  - Random choice of K elements of S
  - Random generation of K elements $x \in R^n$ in the same cuboid as S
  - As above but with a minimum distance between the K elements
- Possible post-processing steps:
  - Merge small clusters
  - Split large clusters, for example if they have a large variance

Clustering Error

- Reduce influence of random initialization with several runs. Choose the best result with respect to the clustering error E_K.
- Often defined as the sum of all quadratic deviations from the mean:

$$m_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

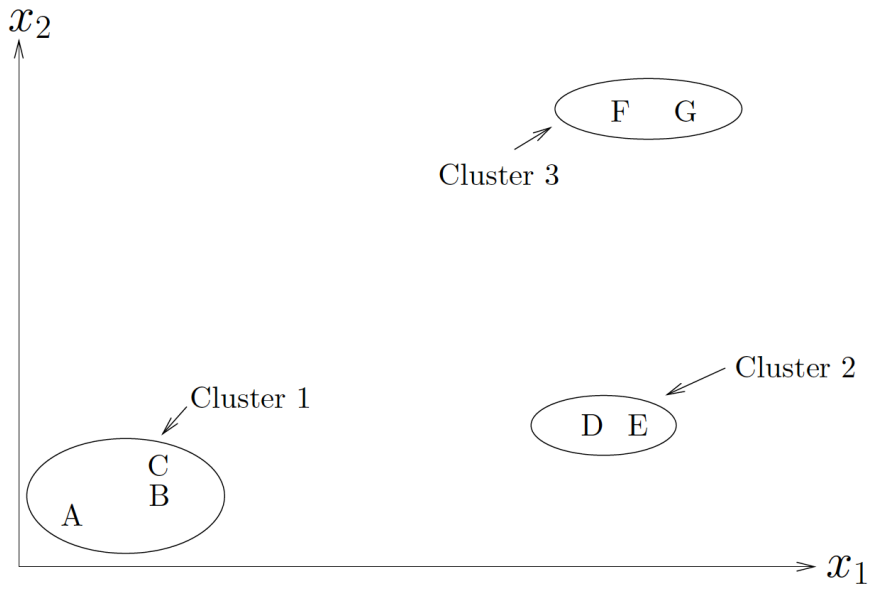
$$e_j = \sum_{x_i \in C_j} (x_i - m_j)'(x_i - m_j)$$

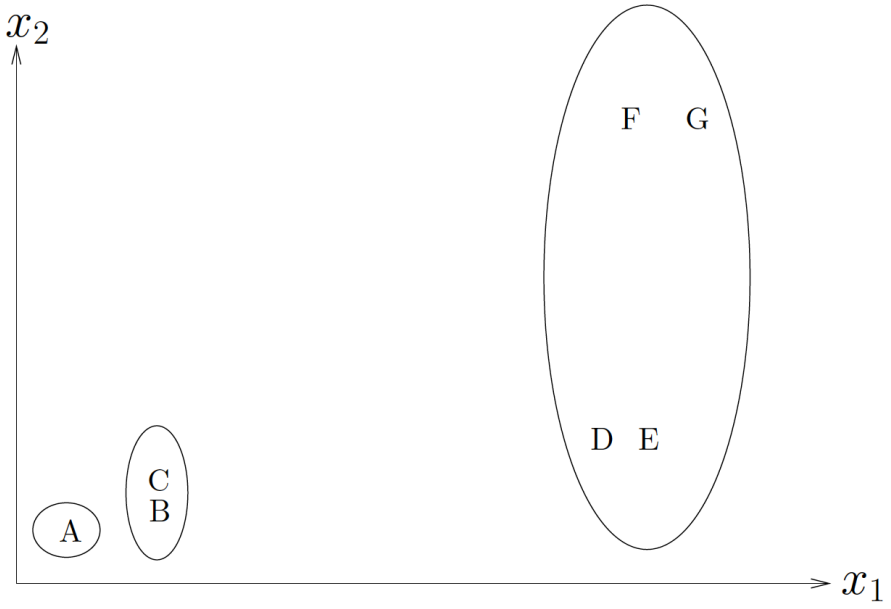
$$E_K = \sum_{j=1}^{K} e_j$$

- E_K stays the same or decreases in each iteration. However, only a local optimum might be found instead of the global optimum.

Example

- Influence of the choice of initial cluster centers.
- {A,D,F} vs {A,B,C}.



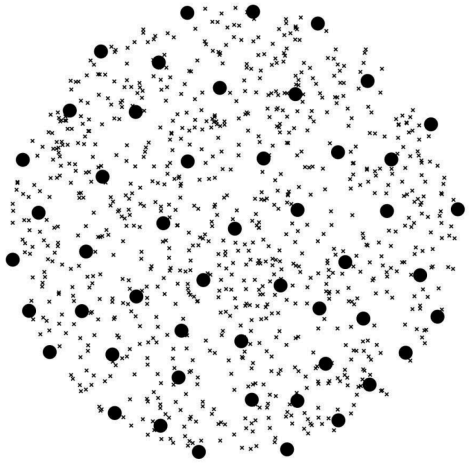


Deterministic K-Means

- To avoid the dependence of K-Means from random initialization, a deterministic choice of initial cluster centers can be used.
- Select for example patterns that are most dissimilar to the previously selected ones, also known as *spanning* selection:

Require: training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; number of clusters K
Ensure: initial cluster centers $M_K = \{m_1, \ldots, m_K\}$

1: $m_1 = \frac{1}{N} \sum_{i=1}^{N} x_i$
2: $M_1 = \{m_1\}$
3: **for** $i = 2$ to K **do**
4: $\quad m_i = \arg\max_{x \in S - M_{i-1}} (\min_{y \in M_{i-1}} ||x - y||)$
5: $\quad M_i = M_{i-1} \cup m_i$
6: **end for**



Global K-Means

- Deterministic K-Means algorithm that increases the number of clusters step-by-step. That is, K solutions with 1,…,K clusters are obtained.
- To compute the next step, each element of S is tested as an additional cluster center for K-Means together with the previous cluster centers.
- Cluster centers are recomputed in each step.

Require: training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; number of clusters K
Ensure: clustering $C = \{C_1, \ldots, C_K\}$

1: $C_1^1 = S$; $m_1 = \frac{1}{N}\sum_{i=1}^{N} x_i$
2: **for** $k = 2$ to K **do**
3: **for** $i = 1$ to N **do**
4: compute $(C_1^k(i), \ldots, C_k^k(i))$ with K-Means$(m_1^{k-1}, \ldots, m_{k-1}^{k-1}, x_i)$
5: **end for**
6: find $(C_1^k, \ldots, C_k^k)$ among $(C_1^k(i), \ldots, C_k^k(i))$ with minimum E_k
7: **end for**
8: return $(C_1^K, \ldots, C_K^K)$

Fast Global K-Means

- Does not test all patterns $x_1,\ldots,x_N$ in Line 4 of the algorithm but instead only one pattern x that maximizes:

$$b = \sum_{j=1}^{N} \max(d_j^{k-1} - \|x - x_j\|^2, 0)$$

- The term d_j^{k-1} is the quadratic Euclidean distance between x_j and the nearest cluster center among $m_1^{k-1},\ldots,m_{k-1}^{k-1}$.
- When computing the next step of Global K-Means, the error reduction is at least b.

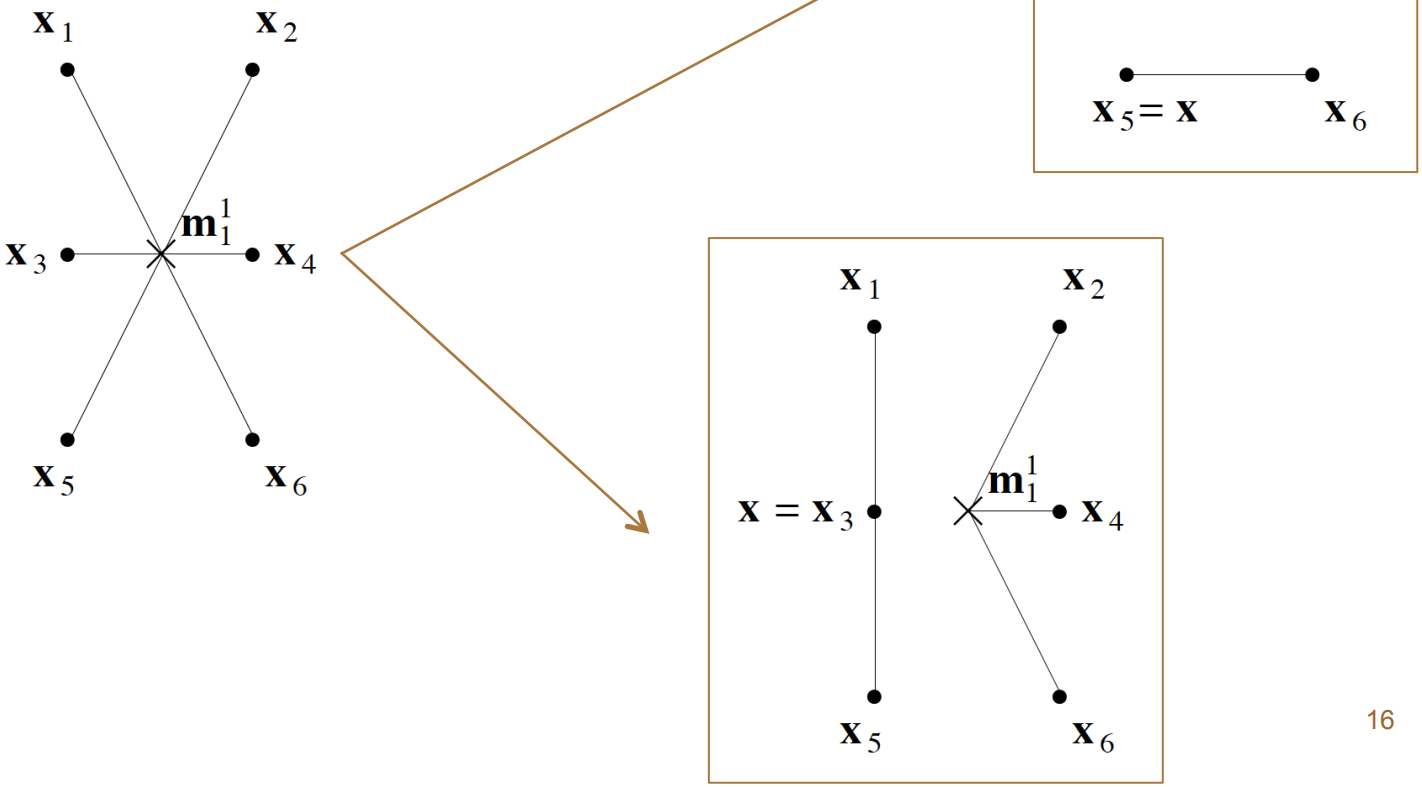
Example

- For x_5, b is:

$$\|m_1 - x_5\|^2 + \|m_1 - x_6\|^2 - \|x_5 - x_6\|^2$$

- For x_3, b' < b is:

$$\|m_1 - x_1\|^2 + \|m_1 - x_3\|^2 + \|m_1 - x_5\|^2 - \|x_3 - x_1\|^2 - \|x_3 - x_5\|^2$$



Graph-Based Clustering

Graph-Based Clustering

- Split set of samples $x_1,\ldots,x_N$ with $x_i \in R^n$ successively into subsets based on a minimum spanning tree (MST).
- *Top-down* divisive approach.
- Time complexity $O(N^2)$ for computing the MST.

Require: training set $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; number of clusters K
Ensure: clustering $C = \{C_1, \ldots, C_K\}$

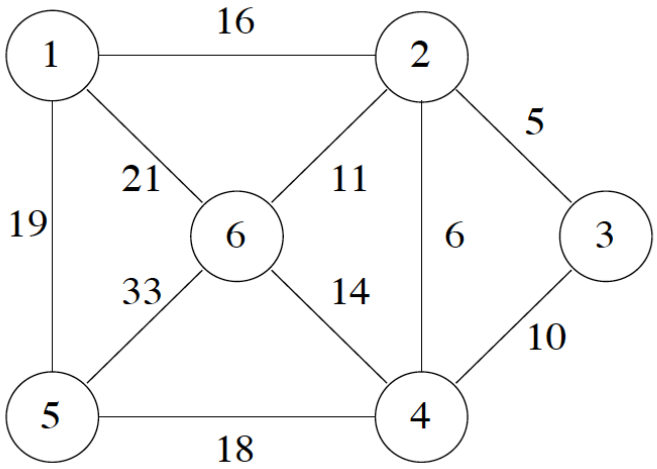
1: compute fully connected graph with edge weights $d(x_i, x_j)$
2: compute the minimum spanning tree (MST)
3: order all MST edges according to their weights (descending order)
4: **repeat**
5: delete the edge with the highest weight
6: **until** number of clusters K is reached

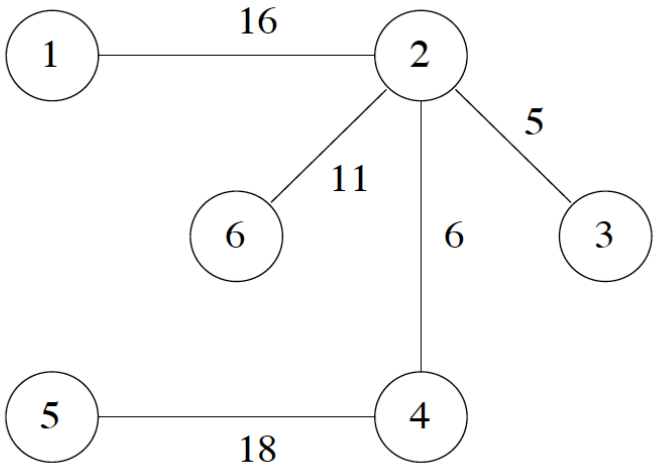
Prim's Algorithm

- Let G=(V,E) be an undirected, connected graph with edge weights w(e)≥0. Then a *spanning tree* of G is an undirected, connected graph T=(V,E') with E' ⊆ E and |E'|=|V|-1.
- A *minimum spanning tree* is a spanning tree with minimum sum of edge weights $\sum_{e \in E'} w(e)$.
- If several edges have the same weight, there might be several MST. Otherwise the MST is unique, as well as the clustering solution.

Require: graph $G = (V, E)$ with weights $w(e)$ for all $e \in E$
Ensure: a minimum spanning tree (MST) of G
1: initialize PMST (partial MST) as an empty tree
2: **repeat**
3: extend the PMST with the minimum-weight edge (plus end node), which connects a node of the PMST with a node outside the PMST
4: **until** PMST has $|V| - 1$ edges
5: MST = PMST

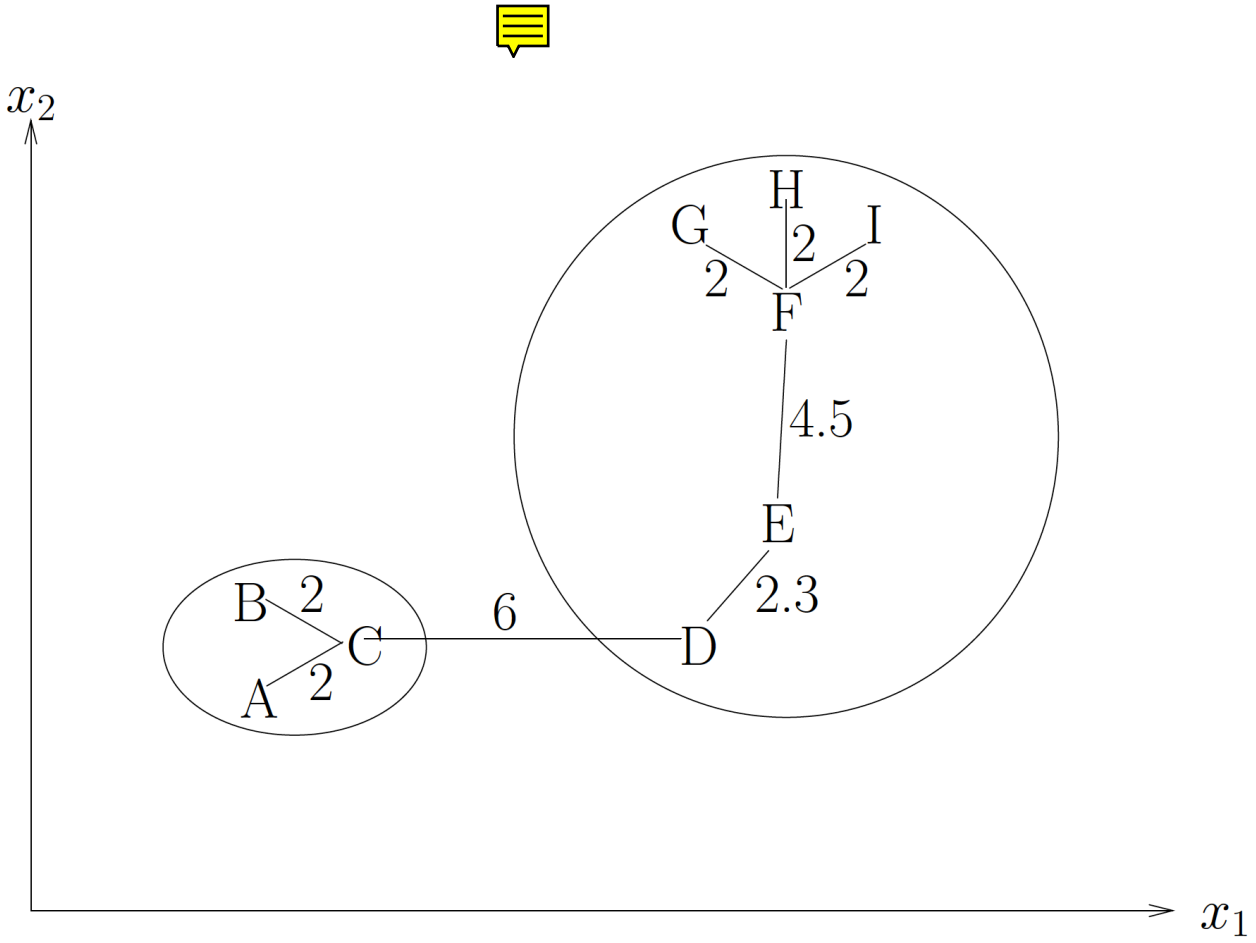
Example 1





repeat-loop	added edge
1	5
2	6
3	11
4	16
5	18

Example 2



Clustering Quality

Clustering Quality

- In the following, four standard measures for the clustering quality are discussed:
  - C-Index
  - Goodman-Kruskal-Index
  - Dunn-Index
  - Davis-Bouldin-Index.
- They are needed for validating and comparing different clustering algorithms.
- They can be used to optimize the number of clusters K.
  - Usually only a range $[K_{min},K_{max}]$ is known beforehand.
  - The clustering algorithms are computed for all $K \in [K_{min},K_{max}]$.
  - The best result is found with respect to the clustering quality.
- Note that there are also other clustering algorithms, like DBSCAN, which do not require K to be known beforehand. However, they typically have parameters as well that can be optimized with respect to the clustering quality.

C-Index

- We define:

$$c(x_i, x_j) = \begin{cases} 1 \text{ if } x_i \text{ and } x_j \text{ belong to the same cluster} \\ 0 \text{ otherwise} \end{cases}$$

$$\Gamma = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d(x_i, x_j) \cdot c(x_i, x_j)$$

$$\alpha = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} c(x_i, x_j)$$

- Γ is the sum of all distances between patterns of the same cluster and α is the number of pairs of patterns that belong to the same cluster.
- Let min be the sum of the α smallest $d(x_i,x_j)$ with $x_i \neq x_j$ and max be the sum of the α largest $d(x_i,x_j)$ with $x_i \neq x_j$. Then the C-Index is defined as:

$$C = \frac{\Gamma - \min}{\max - \min}$$

- Small distances within the same cluster and large distances between different clusters indicate a high clustering quality. Consequently, a good clustering has a small value of $C \in [0,1]$.

Goodman-Kruskal-Index

- We define:

$$\bar{c}(x_i, x_j) = 1 - c(x_i, x_j) = \begin{cases} 1 \text{ if } x_i \text{ and } x_j \text{ belong to different clusters} \\ 0 \text{ otherwise} \end{cases}$$

- A 4-tuple (x_i,x_j,x_r,x_s) with $x_i \neq x_j$, $x_r \neq x_s$, and $(x_i,x_j) \neq (x_r,x_s)$ is *concordant* if:

$$d(x_i, x_j) < d(x_r, x_s) \wedge \bar{c}(x_i, x_j) < \bar{c}(x_r, x_s) \text{ or}$$

$$d(x_i, x_j) > d(x_r, x_s) \wedge \bar{c}(x_i, x_j) > \bar{c}(x_r, x_s)$$

- A 4-tuple (x_i,x_j,x_r,x_s) with $x_i \neq x_j$, $x_r \neq x_s$, and $(x_i,x_j) \neq (x_r,x_s)$ is *discordant* if:

$$d(x_i, x_j) < d(x_r, x_s) \wedge \bar{c}(x_i, x_j) > \bar{c}(x_r, x_s) \text{ or}$$

$$d(x_i, x_j) > d(x_r, x_s) \wedge \bar{c}(x_i, x_j) < \bar{c}(x_r, x_s)$$

- Otherwise, it is neither concordant nor discordant.
- Let S_+ be the number of concordant 4-tuples and S_- be the number of discordant 4-tuples. Then the Goodman-Kruskal-Index is defined as:

$$\gamma = \frac{S_+ - S_-}{S_+ + S_-}$$

- Concordant 4-tuples indicate a high clustering quality. Consequently, a good clustering has a large value of $\gamma \in [-1,1]$.

Dunn-Index

- We consider single-linkage for the cluster distance:

$$d(C_i, C_j) = \min\{d(x,y) \mid x \in C_i, y \in C_j\}$$

- Then the diameter of the cluster is defined as:

$$\Delta(C) = \max\{d(x,y) \mid x, y \in C\}$$

- Let

$$\Delta_{\max} = \max\{\Delta(C_i) \mid 1 \le i \le K\}$$

- Then the Dunn-Index is defined as:

$$D = \min\left\{\frac{d(C_i, C_j)}{\Delta_{\max}} \mid 1 \le i, j \le K\right\} = \frac{1}{\Delta_{\max}} \min\left\{d(C_i, C_j) \mid 1 \le i, j \le K\right\}$$

- A large distance between the two closest clusters and a small maximum diameter among all clusters indicate a high clustering quality. Consequently, a good clustering has large value of $D \in [0, \infty]$.

Davis-Bouldin-Index

- We define:

$$m_i = \frac{1}{|C_i|} \sum_{x_i \in C_i} x_i$$

$$d_i = \frac{1}{|C_i|} \sum_{x_i \in C_i} d(x_i, m_i)$$

$$R_{ij} = R_{ji} = \frac{d_i + d_j}{d(m_i, m_j)}$$

$$R_i = \max\left\{ R_{ij} \mid 1 \le j \le K; i \ne j \right\}$$

- Where R_{ij} measures the compactness of two clusters with respect to their distance. The smaller R_{ij}, the better is the separation of the two clusters. R_i indicates how well C_i is separated from the other clusters in the worst case.
- Then the Davis-Bouldin-Index is defined as:

$$DB = \frac{1}{K} \sum_{i=1}^{K} R_i$$

- A good separability on average indicates a high clustering quality. Consequently, a good clustering has small value of DB$\in[0,\infty]$.

Example

- We consider:

$$R_{12} = \frac{d_1 + d_2}{d_{12}}; R_{13} = \frac{d_1 + d_3}{d_{13}}; R_{23} = \frac{d_2 + d_3}{d_{23}}$$

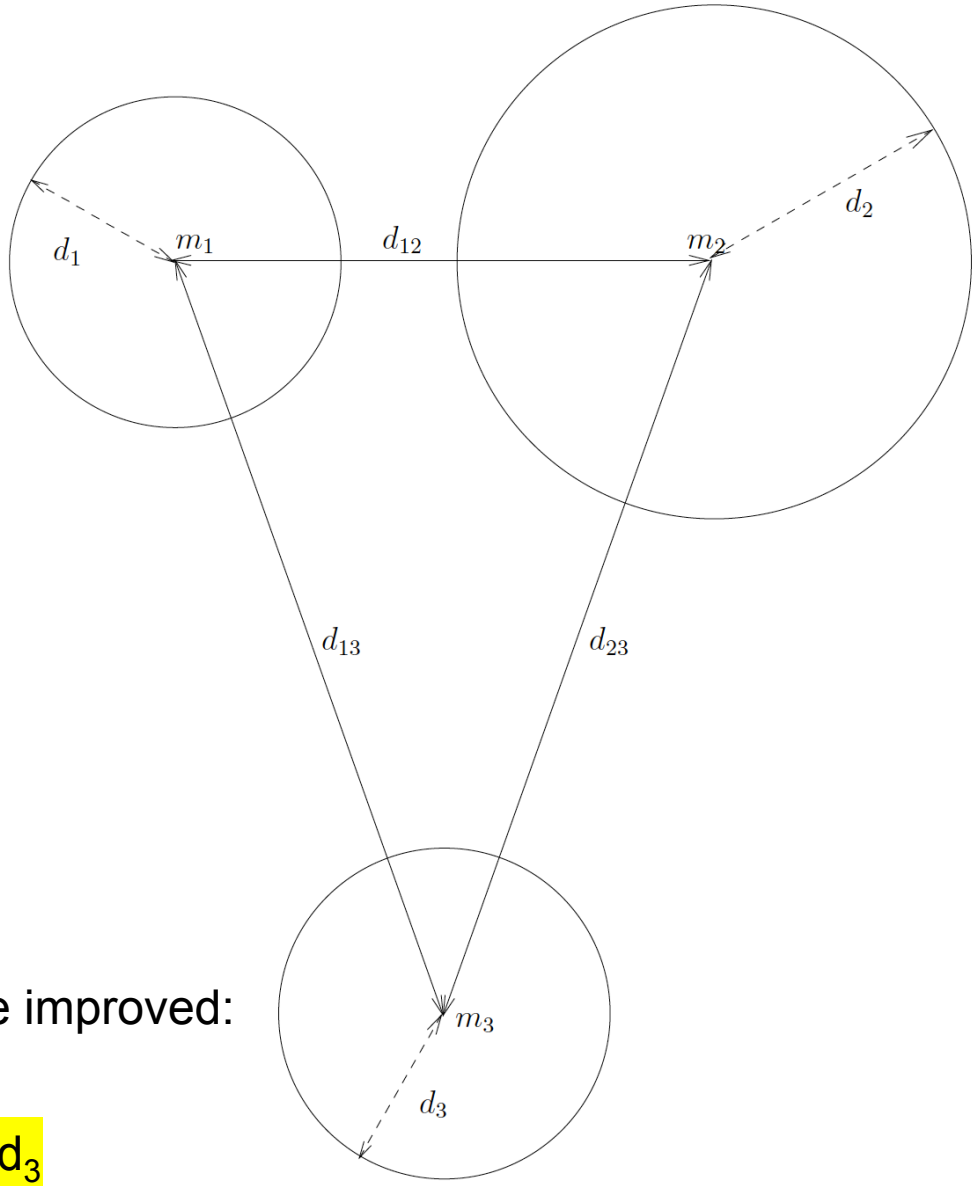
- The worst-case separability:

$$R_1 = R_{12}; R_2 = R_{12}; R_3 = R_{23}$$

- The Davis-Bouldin-Index:

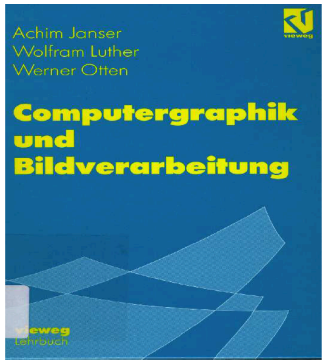
$$DB = \frac{2R_{12} + R_{23}}{3} = \frac{1}{3}\left(\frac{2d_1 + 2d_2}{d_{12}} + \frac{d_2 + d_3}{d_{23}}\right)$$

- That is, the clustering can be improved:
  - By increasing d_{12} or d_{23}
  - By decreasing d_1, d_2, or d_3



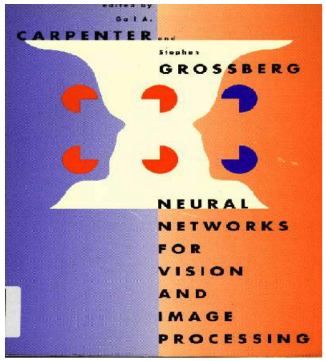
Application Example

- Clustering colors as a pre-processing step for OCR.
- Text elements belong to the same cluster.



colors=151043

clusters=9



colors= 254338

clusters=11

colors=102298

clusters=4